白皮书

借助 ANSIBLE 实现持续集成和交付

简介

Ansible 是一款功能十分强大的开源自动化语言。不同于其他管理工具的是，它还是一款部署和编排工具。Ansible 旨在帮助企业从多方面大幅提升生产效率，轻松应对各种自动化挑战, 并可以为其他自动化解决方案的多项核心功能提供更有效的替代方案， 以及力求解决企业所面临的其他重大 IT 挑战。

这其中的一项关键挑战是如何在不停机的情况下，实现持续集成和持续部署 (CI/CD)。通常，实现这一目标涉及到大量的客户开发、使用多个软件包以及众多企业内部开发产品。Ansible 仅凭单个产品即可轻松提供上述所有功能，且从产品设计之初，就特意围绕这些场景进行了相应的编排。

为什么要实现 CI/CD

过去十年来，软件和 IT 业务实践分析告诉我们，相比更加频繁的、较短发布周期（如“迭代”或“敏捷”开发模式），传统的“瀑布式项目”，不仅开发周期较长，而且开销也要高得多。发布伊始，质保团队就要做好准备，等待执行测试工作，这时候 IT 没有任何需要部署的内容。直到发布周期接近尾声时，IT 才开始进行系统的 QA 检查，此时的开发团队则同时负责两个板块的内容：修补漏洞和进行下一个主要版本的开发规划，并且在这两项工作中不停地频繁切换。

这种做法所面临的关键问题在于，较长的发布周期意味着从发现漏洞到解决漏洞的延时会更久。特别是在涉及数百万用户的大规模网络技术中，这一问题尤显突出。为解决该问题，软件开发行业正在迅速向“早发布，常发布”的方向转变，其口号就是“推进敏捷软件开发”。“早发布，常发布”意味着所有团队成员都能够减少在不同任务间的频繁切换，并在更短时间内完成更改、授权和部署。QA 自动化工程和测试导向开



发 (TDD) 等不同方法则可以增加这些技巧的有效性。如果“早发布，常发布”得以推进，那么 CI/CD 就有可能涅槃重生，这一切似乎就合情合理了。尽管实现敏捷性是一个循序渐进的过程，只要企业能保持朝这个方向发展，终将收获显著成效。因此，实现自动化是达成这一目标的关键。这也使能够支持快速周转、并且仅在必要时才需人工介入的技术日益受到业内的高度关注。

本文将帮助您理解为什么 Ansible 和红帽® Ansible® Tower 是连接您的电脑系统以支持上述流程的理想工具。

为什么要实现零停机?

出现停机和业务中断，不仅会造成企业的收入损失，还会影响客户的服务体验。对于用户遍布全球所有时区的主要网络应用，只有在最严峻复杂的升级过程中才会进行停机处理（当然不包括更新应用版本）。即便是大型企业的内部应用（例如财务或内网系统），关键系统停机也会给生产效率带来重大影响。因此，任何自动化流程均应实现一个目标，即能够以不影响运营能力的方式进行更新。

任何自动化流程均应实现一个目标，即能够以不影响运营能力的方式进行更新。

零停机时间是可以实现的，但需要借助一款强大的多层、多步编排引擎， 如 Ansible 为其提供相应支持。

其他 BUILD 系统

在实现持续交付 (CD) 之前，首先要实现持续集成 (CI)。简单地说，CI 系统是一组 Build 系统，它们可以监控各种源代码控制存储库的变更情况，运行任何适用的测试，并自动构建（理想的情况下还会测试）每个源代码控制变更的最新应用版本，如 Jenkins (jenkins.io)。

顺利实现 CD 的关键则在于，该 build 系统可以在成功构建时即时调用 Ansible。此类 build 的一大优势在于，可以帮助运行单元或集成代码测试的用户快人一步，抢占先机。尽管 Jenkins 可以利用 Ansible Tower 将工件部署到多种环境中，但生产后的 QA/阶段性环境建模可以帮助用户加大胜算，显著提升整个生命周期的可预测性。用户随后还可以参考 Ansible 返回的数据，并直接关联至 build 系统作业中的 Ansible Tower 作业。

事实上，正如我们稍后将详细说明的一样，我们可以在阶段性环境中使用 Ansible Tower，以便在生产部署之前进行测试。



Ansible 独具多层多步编排功能，结合其推送式架构，可以极其迅速地执行这些复杂的工作流程。

滚动更新和多层应用

CD 系统中的绝对要求就是，能够在应用中的不同架构层级间编排滚动更新。Ansible 独具多层、多步编排功能，结合其推送式架构，可以极其迅速地执行这些复杂的工作流程。一层更新后，就可以开始更新下一层，并在各层之间轻松共享数据。

Ansible 支持此操作的一个核心功能就是定义“play”，该功能可精细地选择主机组并为它们分配一些执行任务（或“角色”）。任务中通常会说明某个特定资源所处于的某种特定状态，例如已安装数据包的特定版本，或者检查特定版本的源代码控制存储库。

在大多数网络应用拓扑中，需要按照紧凑的顺序更新特定层级。例如，企业可能首先需要迁移数据库架构并刷新服务器缓存，然后再更新应用服务器。

最重要的是，不要同时管理生产中所有系统的应用和系统配置，这点至关重要。

重新启动服务时，可能无法立即使用。另外，也无法立即更换应用版本。在更新之前，将机器从负载平衡池中移出，这步非常重要。而能够自动将机器放入池中或从中移出，这步也同样关键。

Ansible 允许您通过“Serial (串行)” 关键字控制滚动更新。同时，Ansible 的滚动更新处理还非常智能，如果发生批量故障，滚动更新将会停止，让其他基础架构保持在线。

持续部署管理内容

除了生产服务的持续部署（CD），我们还可以持续部署 Ansible Playbook 内容本身。这允许系统管理员和开发人员将其 Ansible Playbook 代码提交给中央存储库，在阶段性环境中运行一些测试，以及在通过这一阶段后自动将那些配置应用到生产中。这可以将部署软件的相同软件流程应用于配置管理，从而获得更多相同的优势。

由于软件或配置更改往往是造成意外中断的一个重要原因，这样使得自动测试和人工检查成为十分有用的实施方法。同时配合使用代码检查系统，如 Gerrit (gerritcodereview.com)，要求在应用更改前退出多用户模式。



支持负载平衡和监控的滚动更新

Ansible 擅长在执行滚动更新期间应用负载平衡器等工具。在任何 Playbook“host loop”过程中，都可以执行类似操作，比如“代表主机 Y 在系统 X 中执行此操作”。

这包括与各种负载平衡器进行通信，以及标记特定主机的中断窗口，以便禁用当前正在更新的主机的监控提醒。“禁用监控 - 从池中移出 - 更新层级 - 添加至池 - 启动监控”（disable monitoring - remove from pool - update tier - add to pool - enable monitoring）等简单习语可以帮助用户实现无停机更新操作，也不会在无人工干预的情况下自动执行假传呼警报。

将 ANSIBLE 与测试阶段集成

使用包含多个存储库资源（inventory sources）的 Ansible Tower，可以帮助用户在测试环境中检测 Ansible Playbook，并在提交至生产环境前即实现成功更新。在真实环境下，此设置非常适合在更新系统前，先在小范围内进行生产环境更新场景的模拟构建。用户只需对 Ansible Playbook 使用“-i”参数来指定 Playbook 应该在什么存储库资源上运行，并在测试和生产阶段中使用相同的 Playbook 即可。

基于版本控制进行部署

各家企业都喜欢将他们的应用与操作系统软件包（RPM、debs 等）一起打包，但在多种情况下，特别是对于动态网络应用，这种打包并无必要。为此，Ansible 提供了多种模块，可直接从版本控制进行部署。Playbook 可以要求检查特定标签或版本的存储库，然后系统将确保其在各个服务器间都保持相应的正确版本。需要更改软件版本时，Ansible 还可以报告更改事件，以触发其他后续操作，从而禁止不必要地相关服务重启。

与监控工具集成

由于 Ansible 是一个编排系统，所以也可以使用 APM 监控工具进行集成。例如，我们假设在集成测试或部署期间，APM 系统需要在应用中安装/更新一个代理。Ansible 可以设置一个角色来处理应用堆栈的代理安装。当代理在线后，Ansible 可以在监控堆栈中设置警报（如果尚未设置）。之后，这种监控功能将向相关的负责团队发回即时数据；让他们知道应用是否正常运行。

升级后，如果在生产中应用出现问题，监控工具就会调用 Ansible 恢复到之前构建良好的应用。当然，只有在应用允许恢复构建的情况下才可以执行此操作。



回调和插入功能

在 CI/CD 环境中，在事件发生时提供警报通常会很有帮助。Ansible 包含多个执行设施，包括一个集成邮件模块。此外，它的回调功能还允许在任意系统中插入通知，包括聊天广播工具（IRC、Campfire 等）、自定义记录或内部社交媒体。

适用于部署的所需状态资源模型

之所以 Ansible 对于软件部署十分有帮助，其中一个关键功能就是其使用了基于状态的资源模型，这在更新软件时配置管理工具非常有用。与那些通常还需要部署额外软件和脚本的开源工具不同，Ansible 无需这些操作。

这也让 Ansible 能够精密控制不同系统层次间的事件顺序，将操作委派给其他系统，还可以在相同流程中混合使用资源模式指令（例如设定“数据包 X 的状态为 Y”）和传统命令模式（“run script.sh”）。

通过将配置和部署统一到一个工具链下，可以减少管理不同工具的开销，还能在操作系统和应用策略之间实现更好的统一。

Ansible 还可以轻松执行不同条件下的测试命令，然后根据那些命令的结果制定符合条件的决策。通过将配置和部署统一到同一个工具链下，不仅可以减少企业管理不同工具的开销，还能够在操作系统和应用策略之间实现更好的统一。

将测试嵌入到部署中

功能强大，同时责任重大。设置自动化 CD 系统的一项风险在于，不良配置可能会传播到所有相关系统中。为有效应对这一情况，Ansible 可以直接在 Playbooks 中添加测试，从而在出现问题时中止滚动更新。针对不同情况，Ansible 可以检查用“命令(Command)”或“脚本(Script)”模块部署，或作为本地 Ansible 模块部署，以对不同状态进行测试。这可能包括服务的功能状态。

无论何时，使用“故障（fail）”模块都会中止主机执行。因此，用户可以轻松地在滚动更新窗口中尽早捕捉错误。例如，假设在测试环境和生产之间的差异造成了某个配置错误，从而导致了生产环境内的部署中断。这时，可以写入 Ansible Playbook，以便在滚动更新窗口的第一阶段立即停止更新流程。如果有 100 个服务器并且当前更新窗口为 10，那么更新窗口就会停止运行并允许干预。错误纠正后，只需要重新运行 Playbook 以继续更新即可。

发生故障时，Ansible 不会继续运行，从而导致系统配置不全。它会提醒您发生错误以便进行及时处理，并提供相应的报告摘要，帮助您了解更新周期中，哪些主机出了问题，以及每个平台上执行了多少更改等信息。



合规测试

在许多环境中，除非一些确实必要的更改，否则不会应用配置更改。在“按下按钮”之前，这些更改应先被有关人员了解，并获得相应更改许可。这种情况下，CD 系统仍然可以为我们提供一些有益的提示信息。

Ansible 包含的一个 dry-run 模式，可以通过“--check”标记，报告运行 Playbook 流程将完成哪些更改。

这时，Ansible 包含的一个 dry-run 模式，可以通过“--check”标记报告运行 Playbook 流程可以完成哪些更改。由于这会跳过实际命令执行且对不考虑潜在的故障脚本，所以这只是一项准则。同时它也是工具包中的实用工具。

即便在有新的构建产品出现时进行连续部署，用户也依然可以进行频繁地合规性检查。当生产中的产品可能因人工干预而发生变更，并且需要通过运行其他 Playbook 进行纠正时，这一功能还可以为用户提交相应的检查报告，同时也为用户检测是否需要进行软件变更，是否需要纠正权限设置等工作提供了便捷方式。

实现部署自动化

Ansible 工具丰富，功能强大，是一款理想的 CI/CD 解决方案。这些功能包括，能够在零停机条件下，在滚动更新工作流程中精密编排多层、多步流程。这正是通过 Ansible 的独特架构以及无代理系统（这一系统即可增加安全性，且无需管理流程）来实现，它可以简单的方式呈现可人工读取的结果，而这些工作以往需要大量复杂的人工 IT 流程才能实现。如今，那种将运营团队集中在一个会议室里，密集执行复杂的人工和自动化操作的时代已经一去不复返。Ansible 正在帮助企业实现全自动化部署。

“全自动”意味着要能支持真正的敏捷流程：更快提供变更、更少错误、更少的工作职能转换。对于任何核心的企业内部 IT 服务或高流量的网络资源而言，消除业务中断状况，特别是人为更改错误而导致的状况，都是至关重要的。

借助 Ansible 的强大架构功能，以及与 Jenkins 等 CI 系统紧密集成等功能，不仅可以帮助企业实现自动化配置，还可以实现 IT 流程的全方位自动化控制。这正是我们将 Ansible 当做一款编排引擎看待，而不仅仅是一个配置管理或软件部署工具的原因。

示例和补充信息

您可访问以下网址，查找 Ansible 的一些相关基本示例：
github.com/ansible/ansible-examples

立即使用 Ansible Tower：
ansible.com/tower



关于红帽 ANSIBLE TOWER

Ansible 是由红帽公司赞助的一个开源社区项目，也是企业实现 IT 自动化的最简化解决方案。Ansible 是唯一一款适用于整个 IT 团队——从系统到网络管理员、再到开发人员和经理的自动化语言。红帽® Ansible® Automation 提供可自动执行整个应用生命周期（从服务器到云到容器以及之间的所有设施）的企业就绪型解决方案。红帽® Ansible® Tower 作为一款商业产品，旨在通过对 Ansible 驱动的环境提供资源管控，知识库以及用户授权，帮助团队有效管理复杂的多层部署。

关于红帽

红帽是世界领先的开源软件解决方案供应商，依托社区力量为客户提供稳定可靠且高性能的云技术、Linux、中间件、存储和虚拟化技术。红帽还提供屡获殊荣的专业支持、培训和咨询服务。作为紧密连接全球企业、合作伙伴和开源社区的中心，红帽致力于通过为广大客户提供实用、创新型技术产品，有效释放其宝贵资源以推动业务增长，并为未来 IT 发展奠定坚实基础。

北美地区	欧洲、中东及非洲	亚太地区	拉丁美洲地区
1 888 REDHAT1	00800 7334 2835	+65 6490 4200	+54 11 4329 7300
www.redhat.com/zh	europe@redhat.com	apac@redhat.com	info-latam@redhat.com

